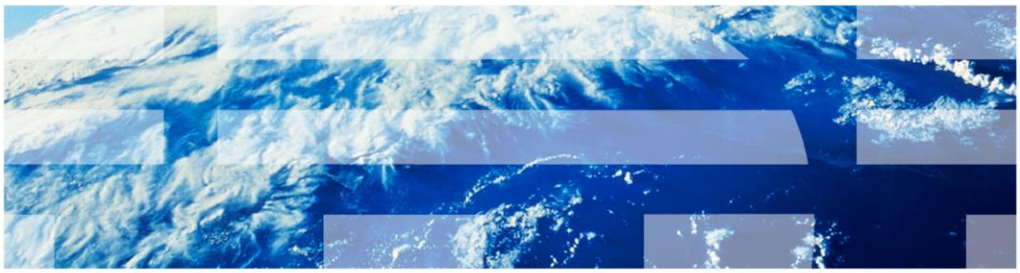
IBM Workload Deployer V3.1

System plug-ins and pattern types

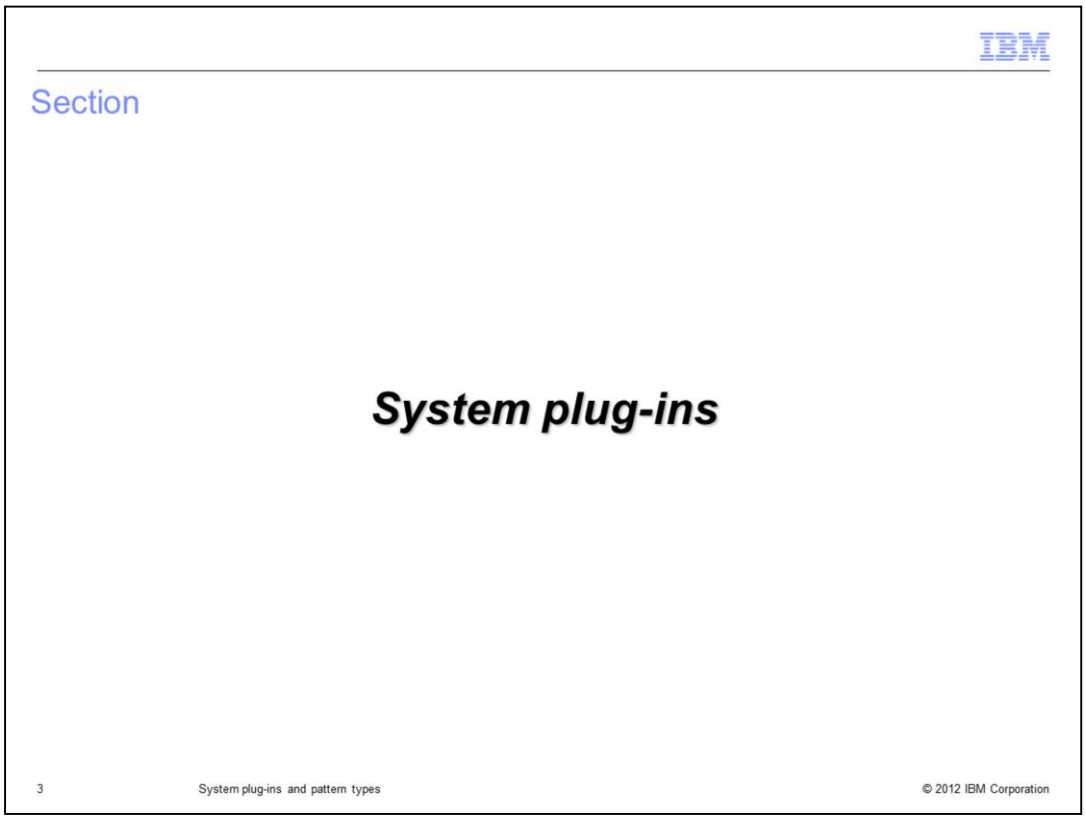


This presentation will cover IBM Workload Deployer V3.1 system plug-ins and pattern types.

IBM

Table of contents

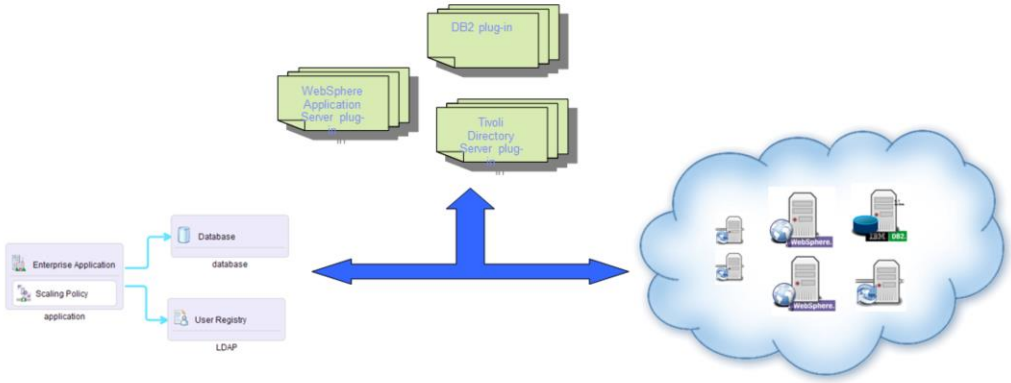
The presentation will provide an overview of system plug-ins and pattern types; then show plug-ins that must be configured before they can be used.



This section covers system plug-ins.

What a system plug-in is

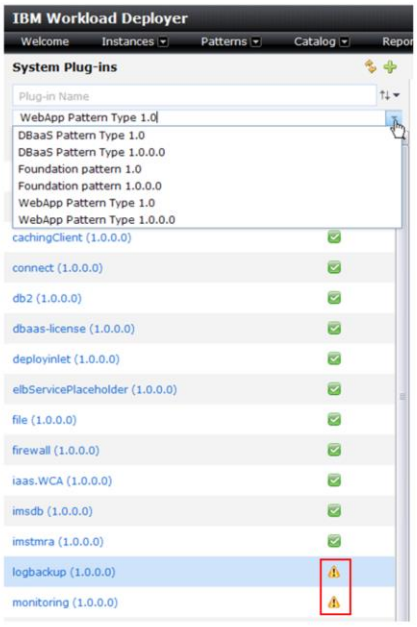
- System plug-ins control the end-to-end handling of a particular capability
  - Component, link and policy definitions
  - Configurable properties
  - Provide middleware implementation at deployment
  - Activation



A system plug-in controls the end to end handling of a particular capability. When building your virtual application or database pattern the plug-in defines the available components, supported linkages and policies and configurable properties for each. At deployment time the plug-in provides implementation details and can further augment the deployed foundation image.

System plug-in filter

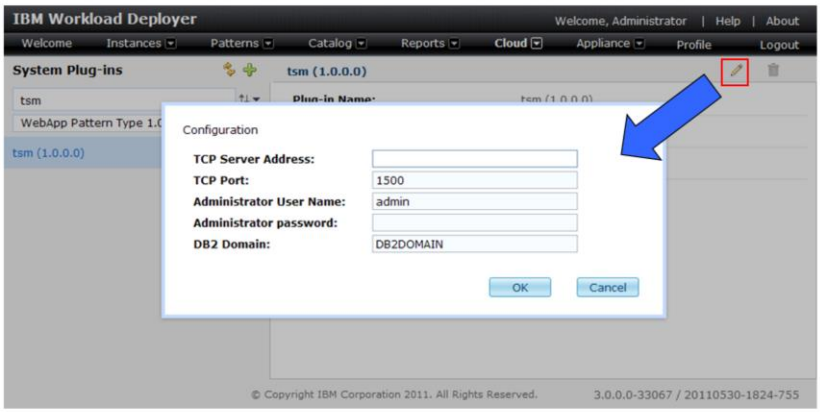
- From system plug-ins panel you can
  - Filter plug-ins by pattern type
  - Filter plug-ins by system plug-in name
  - Review and update system plug-ins
- Incomplete configuration represented by yellow exclamation point



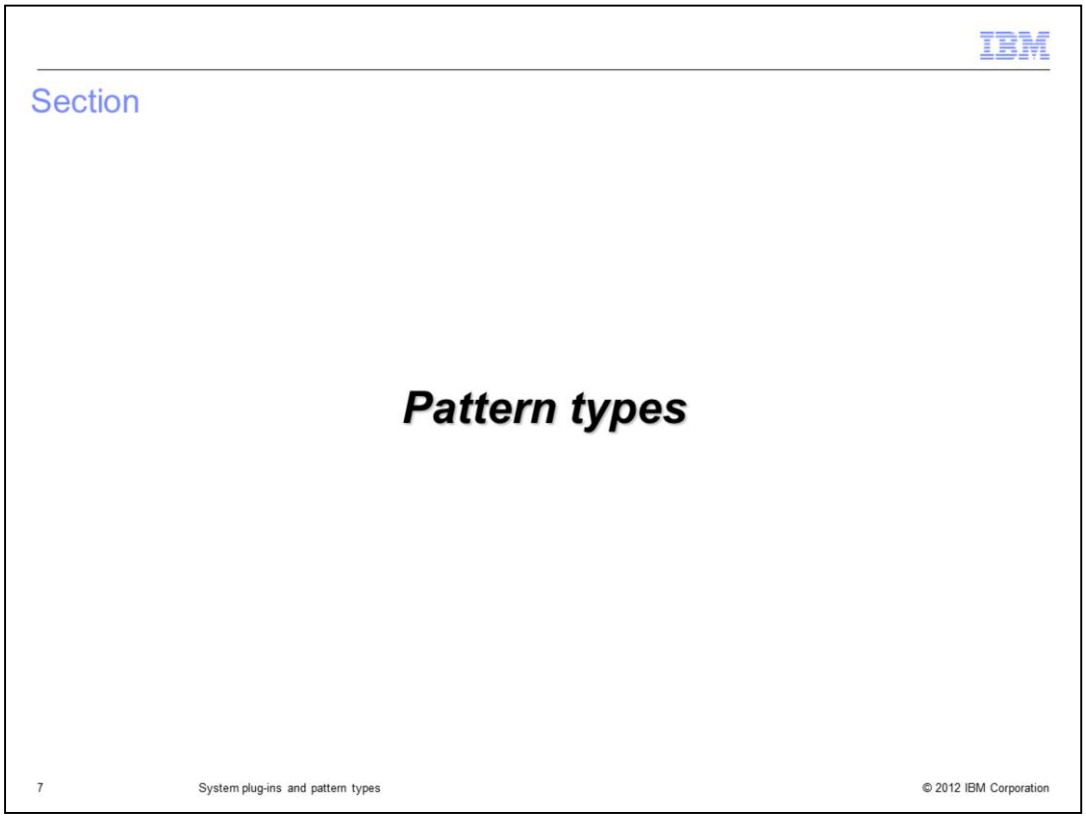
To view and update available system plug-ins navigate to Cloud > System plug-ins. From this view you can filter plug-ins by name and pattern type. A yellow triangle with an exclamation point means that the system plug-in's configuration is incomplete. You must complete a system plug-in's configuration before use.

System plug-in configuration

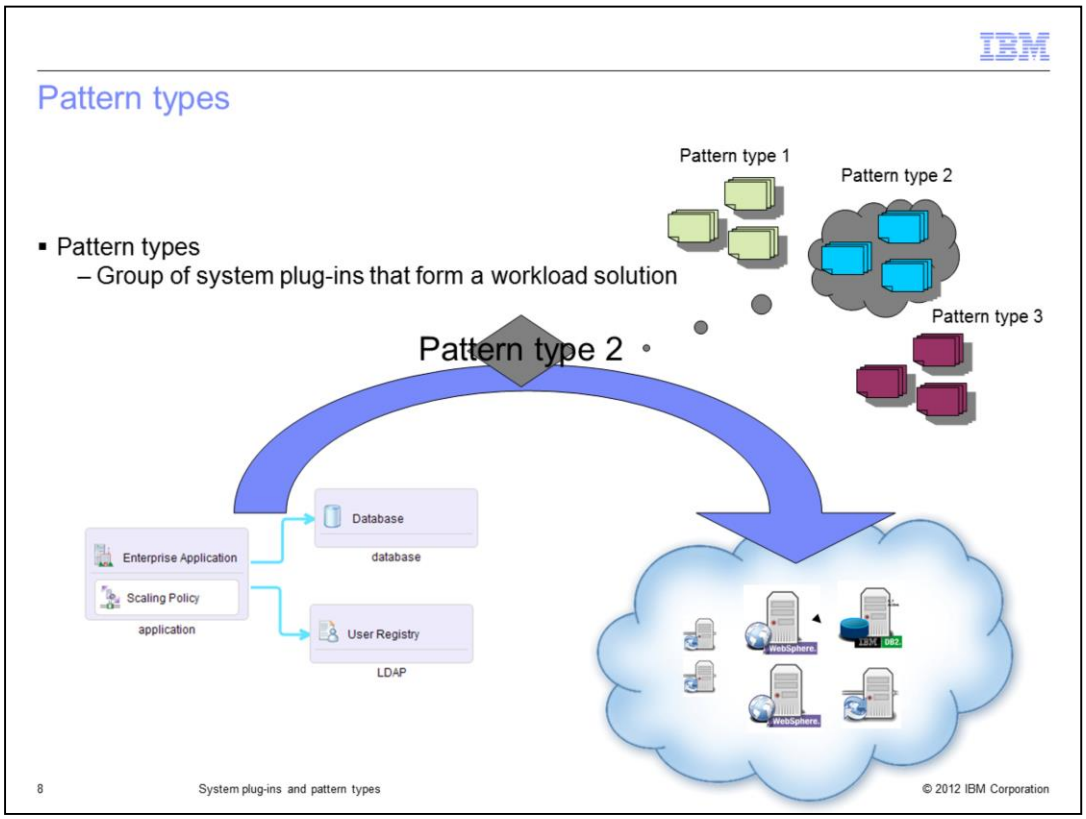
- Some system plug-ins allow or even require additional configuration before use
  - For a complete list visit the IBM Workload Deployer information center



Some system plug-ins allow, or even require, additional configuration before use. Each system plug-in's configuration points are unique to that system plug-in. To configure a system plug-in click the edit button to bring up the available configuration points and enter the required information.



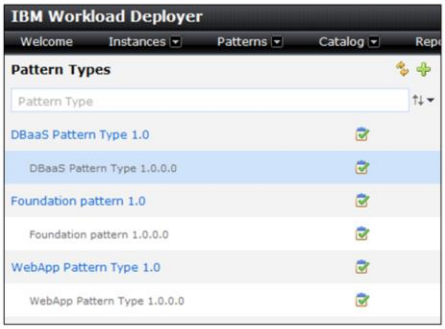
This section covers pattern types.



A pattern type groups a set of related system plug-ins into what is called a workload solution. When you build out your virtual application you choose a solution; not individual system plug-ins.

Pattern type filter

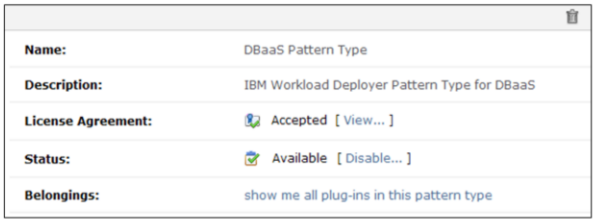
- From the pattern types panel you can
  - Filter pattern types by name
  - Review and update pattern types



To view and update available pattern types navigate to Cloud > Pattern types. From this view you can filter pattern types by name.

Pattern type configuration

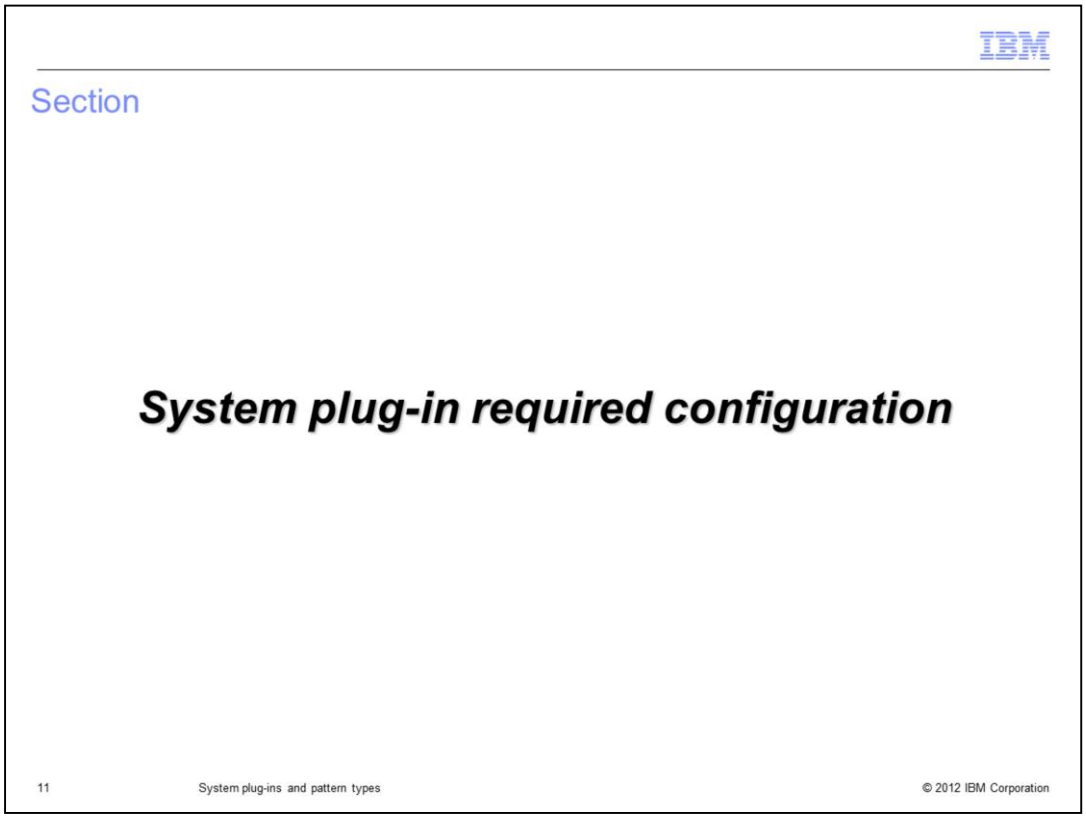
- Before a pattern type can be used these steps must be complete
  - Accept license
  - Enable
- Review all system plug-ins that are associated with a pattern type



To view or update a pattern type's configuration click the pattern type to bring up its configuration in panel in the administrative console's details frame. From here you can view the pattern type description, license, status and associated plug-ins. Before you can use a pattern type you must accept the license and enable the pattern. If you do not perform these two steps the offending pattern type and all associated system plug-ins are unavailable. Separating the license acceptance and enablement steps allows for two scenarios:

Scenario one: accept license, configure system plug-ins and then enable the pattern type. This ensures that the pattern type is completely configured before it is used.

Scenario two: disable pattern type to prevent further deployments. Once all existing deployments using the pattern type are stopped you can remove the pattern type. This scenario allows you to clean the appliance of previous updates.



This section covers system plug-ins that require configuration.

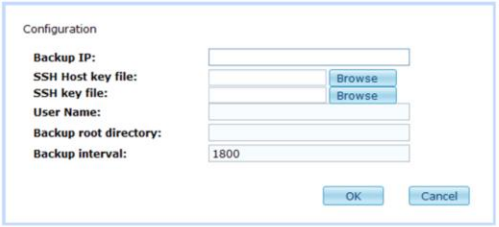
System plug-in required configuration

- Some system plug-ins allow you to further configure them before use
- Some system plug-ins require configuration before use
  - Foundation Pattern Type plug-ins
    - logbackup
    - monitoring
  - WebApp Pattern plug-ins
    - wasctg
    - wasoracle
  - DBaaS Pattern Type plug-ins
    - tsm

Workload Deployer V3.1 comes preloaded with three pattern types: Foundation Pattern Type, WebApp Pattern Type and DataBase as a Service Pattern Type. Each of these pattern types includes a set of system plug-ins. Many of these system plug-ins allow further configuration before use. Some require configuration before they can be used.

Foundation pattern type plug-ins

- logbackup
  - Backup at regularly scheduled intervals



- monitoring
  - Integrate with existing IBM Tivoli® Monitoring server

13 System plug-ins and pattern types

The Foundation Pattern Type includes two system plug-ins that require configuration before use.

The "**logbackup**" plug-in, when configured, will result in automatic backups of the virtual application's logs at regularly scheduled intervals. The logs are placed in the specified SSH storage location.

The "**monitoring**" plug-in allows you to integrate the Tivoli Monitoring agents that are automatically deployed by Workload Deployer with an existing IBM Tivoli Monitoring server.

DBaaS pattern type plug-ins

- tsm
  - Database backup to IBM Tivoli Storage Manager server

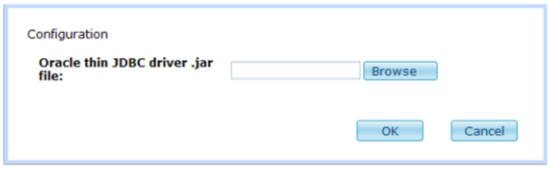
The DataBase as a Service Pattern Type includes one system plug-in requiring configuration before use.

The "**tsm**" plug-in, when configured, allows you to initiate backups of your Workload Deployer deployed databases to an existing IBM Tivoli Storage Manager server.

WebApp pattern type plug-ins

- wasctg
  - Resource adapter to connect to an existing CICS® server

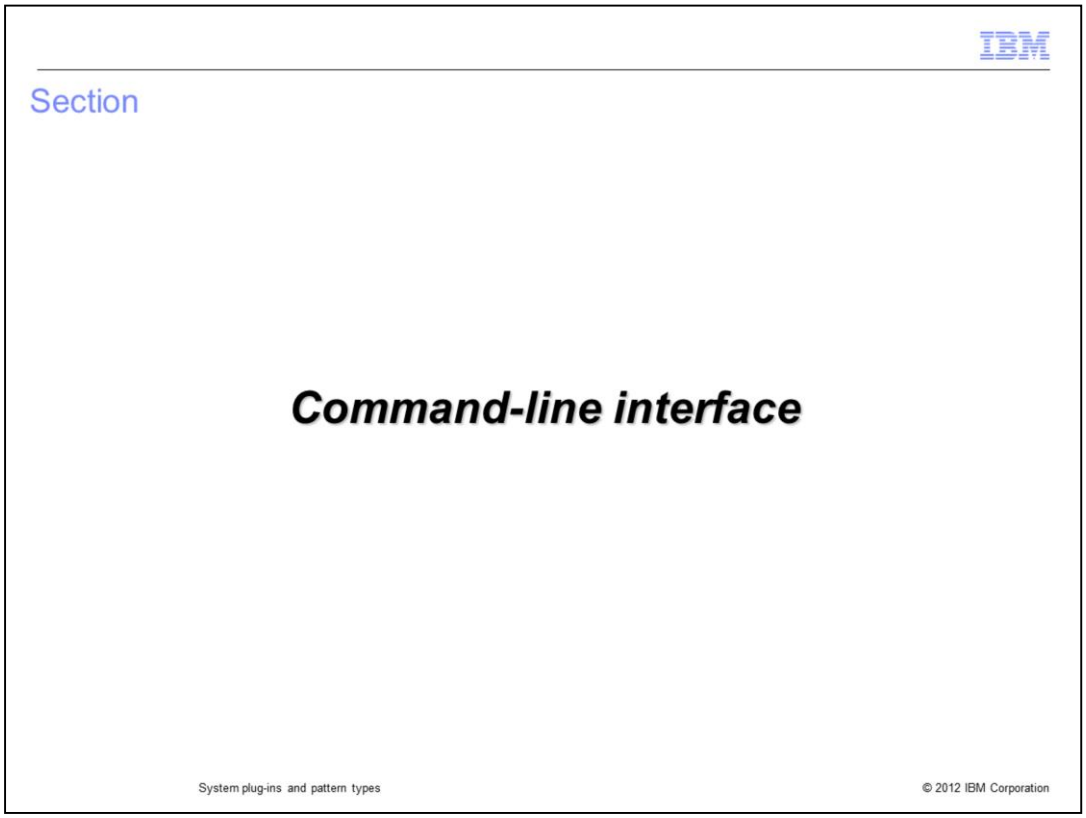
- wasoracle
  - Driver to connect to an existing Oracle database



The WebApp Pattern Type includes two system plug-ins that require configuration before use.

The "**wasctg**" plug-in, when configured, enables the "Existing CICS Transaction Gateway" component, allowing you to connect to an existing CICS server.

The plug-in "**wasoracle**" plug-in, when configured, enables the "Existing Database (Oracle)" component allowing you to connect to an existing Oracle Database.



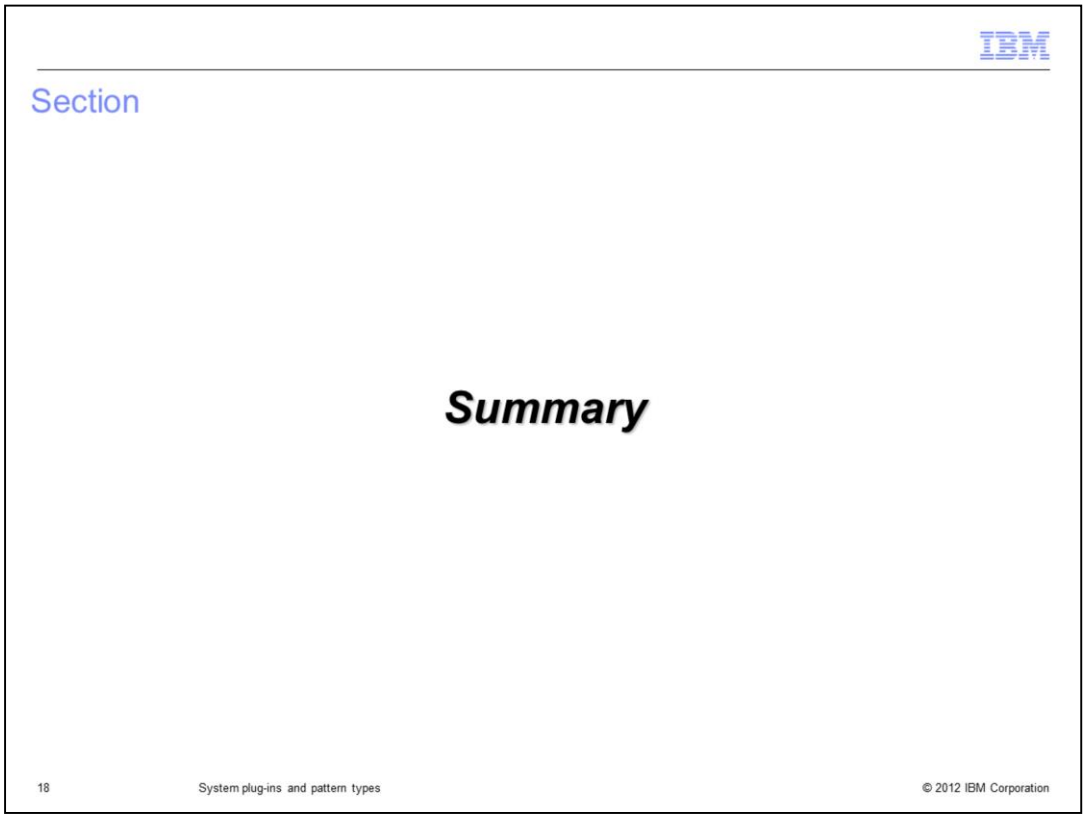
This section covers the command-line interface.

System plug-ins

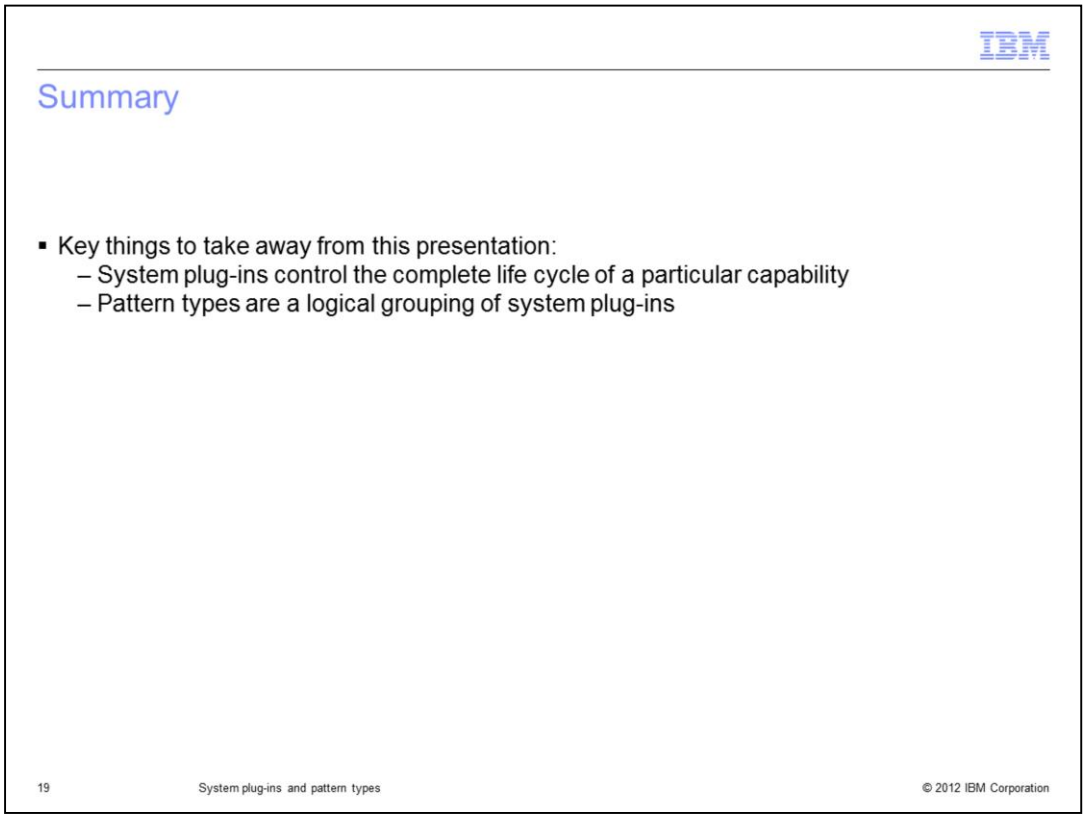
- Subset of available system plug-in command-line interface support:
  - Search and list available system plug-ins
  - Create and delete system plug-ins
  - Get configuration details

```
>>> deployer.plugins
[
  {
    "access_rights": (nested object),
    "content_md5": "90A4CDFE8385B856A130AD4BF2F18D02",
    "content_type": "application/json",
    "create_time": "2011-05-20T02:33:10Z",
    "creator": "cbadmin",
    "last_modified": "2011-05-20T02:33:10Z",
    "last_modifier": "cbadmin",
    "name": "wasvmqq/1.0.0.0"
  },
  {
    "access_rights": (nested object),
    "content_md5": "4E7EBF48824CA6A3C4AC21AA6FE3FA04",
    "content_type": "application/json",
    "create_time": "2011-05-20T02:33:26Z",
    "creator": "cbadmin",
    "last_modified": "2011-05-20T02:33:26Z",
    "last_modifier": "cbadmin",
    "name": "webservice/1.0.0.0"
  },
  {
    "access_rights": (nested object),
    "content_md5": "B091C6AF4873A5AFAAFAE5AF31BBA1CC",
```


The command-line interface provides support for system plug-ins. In the command-line interface you can: search and list available system plug-ins; Create and delete system plug-ins; and get configuration details of a specific system plug-in.



This section will give a brief summary.



This presentation covered pattern types and system plug-ins. The keys to remember are:

System plug-ins define and control the complete life cycle of a particular capability.

Pattern types are logical groups of system plug-ins.

Virtual applications are backed by a pattern type and not any one particular system plug-in.

Trademarks, disclaimer, and copyright information